13 March 2024

Open Confidential Computing Conference

Evolution of the Arm Confidential Compute Architecture

And how Arm is supporting ecosystem developers

Gareth Stockwell

Senior Principal
Systems Architect

arm

Nick Sample

Senior Manager,
Education Engagements

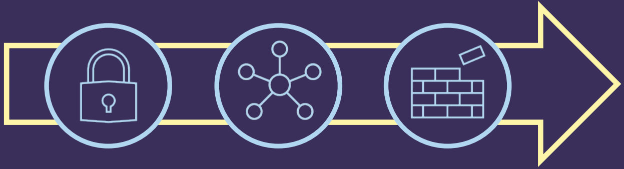
arm

Paul Howard

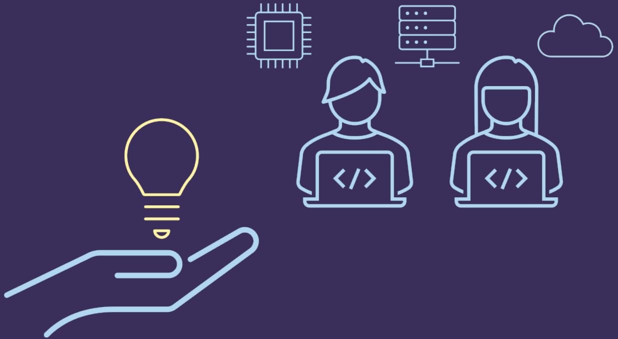
Principal System
Solutions Architect

arm

Agenda



Evolution of the Arm CCA Platform

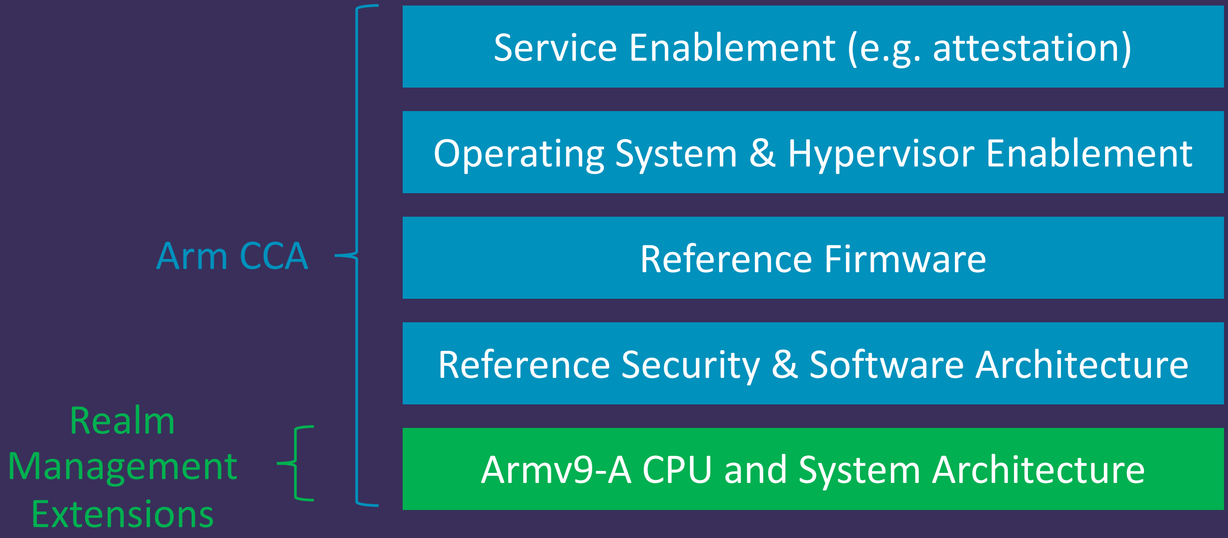


How Arm Is Supporting Ecosystem Developers

Arm CCA overview

Gareth Stockwell

Components of Arm CCA



Service Enablement (e.g. attestation)

Operating System & Hypervisor Enablement

Reference Firmware

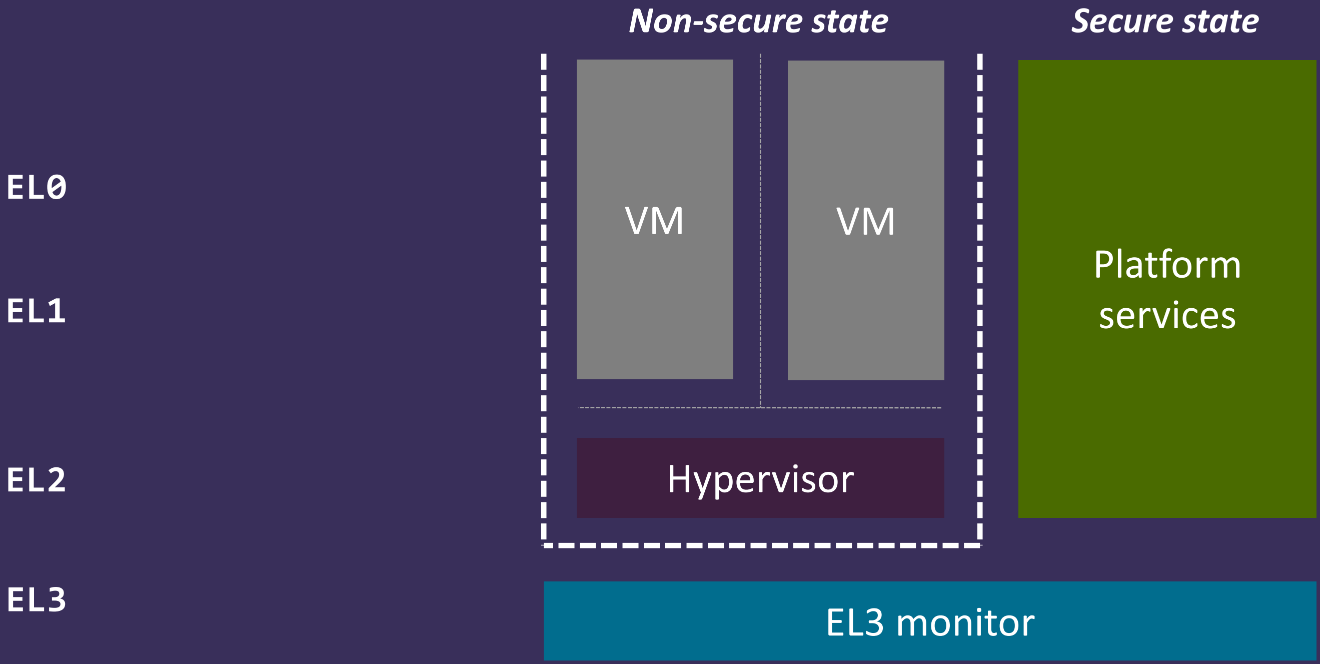
Reference Security & Software Architecture

Armv9-A CPU and System Architecture

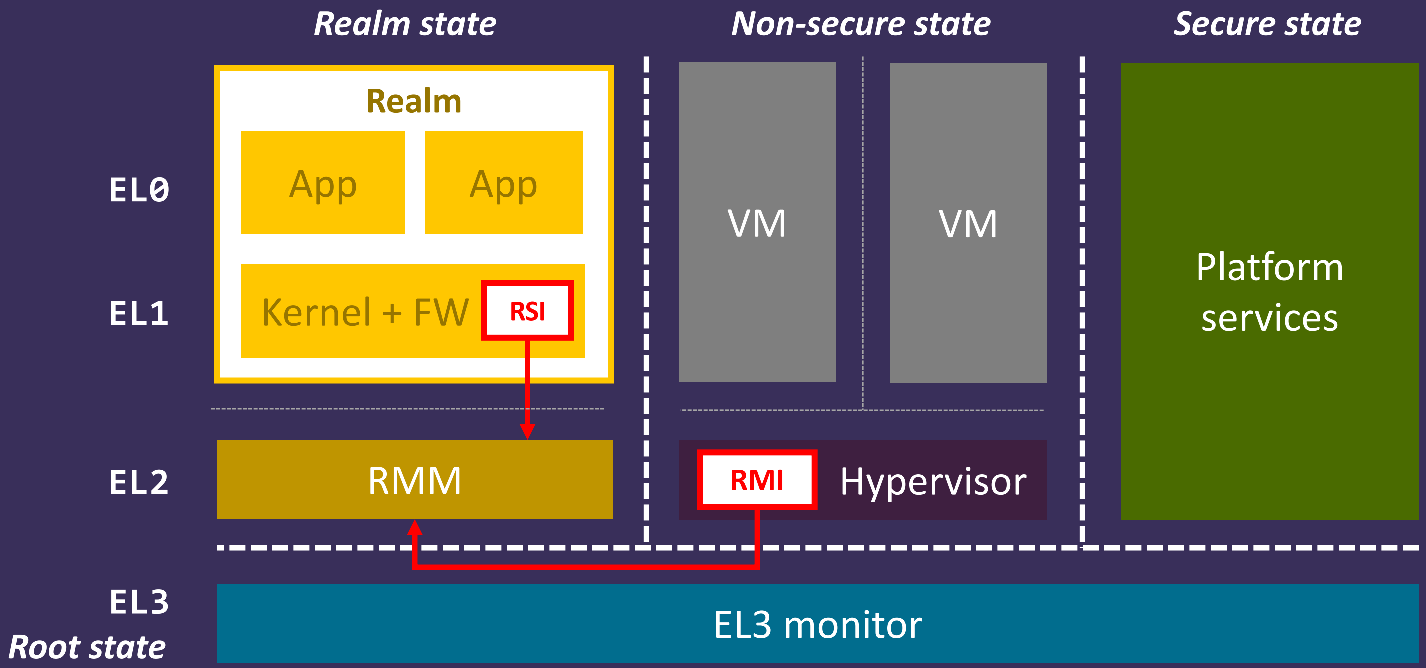
Arm CCA

Realm Management Extensions

Arm pre-CCA software architecture



Arm CCA software architecture



Resources publicly available today

- Arm Architecture Reference Manual for A-profile architecture (includes RME)
- RME system architecture specification
- System MMU architecture specification

- Armv8-A Base Architecture Fixed Virtual Platform (FVP) (implements RME)
- Arm Neoverse Freemont Reference Design FVP (implements RME)

- Realm Management Monitor v1.0 specification (firmware interfaces)

- Reference code (TF-A, TF-RMM) and RFC patches (Linux, EDK2)

- CCA learning resources including how to create and run a Realm on the FVP

Evolution of Arm CCA

Gareth Stockwell

Evolution of Arm CCA

Further strengthen security guarantees provided to end users

Provide feature parity between Realms and non-confidential VMs

Provide additional flexibility to Arm CCA platform owners

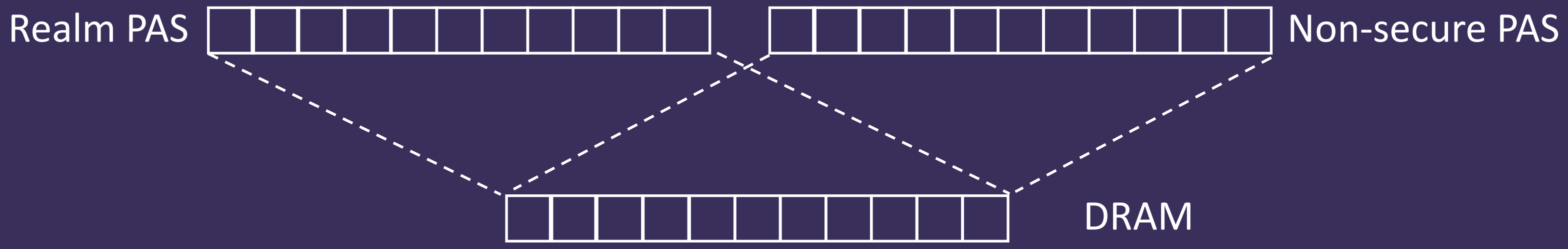
Evolution of Arm CCA

Further strengthen security guarantees provided to end users

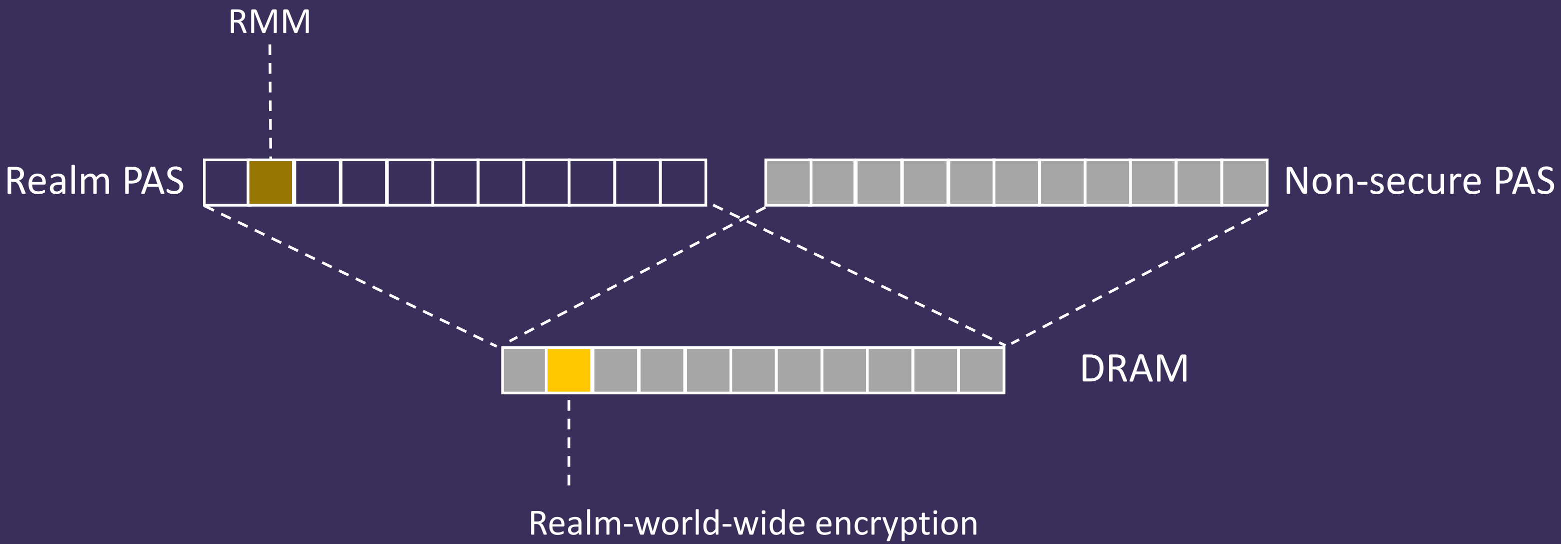
Provide feature parity between Realms and non-confidential VMs

Provide additional flexibility to Arm CCA platform owners

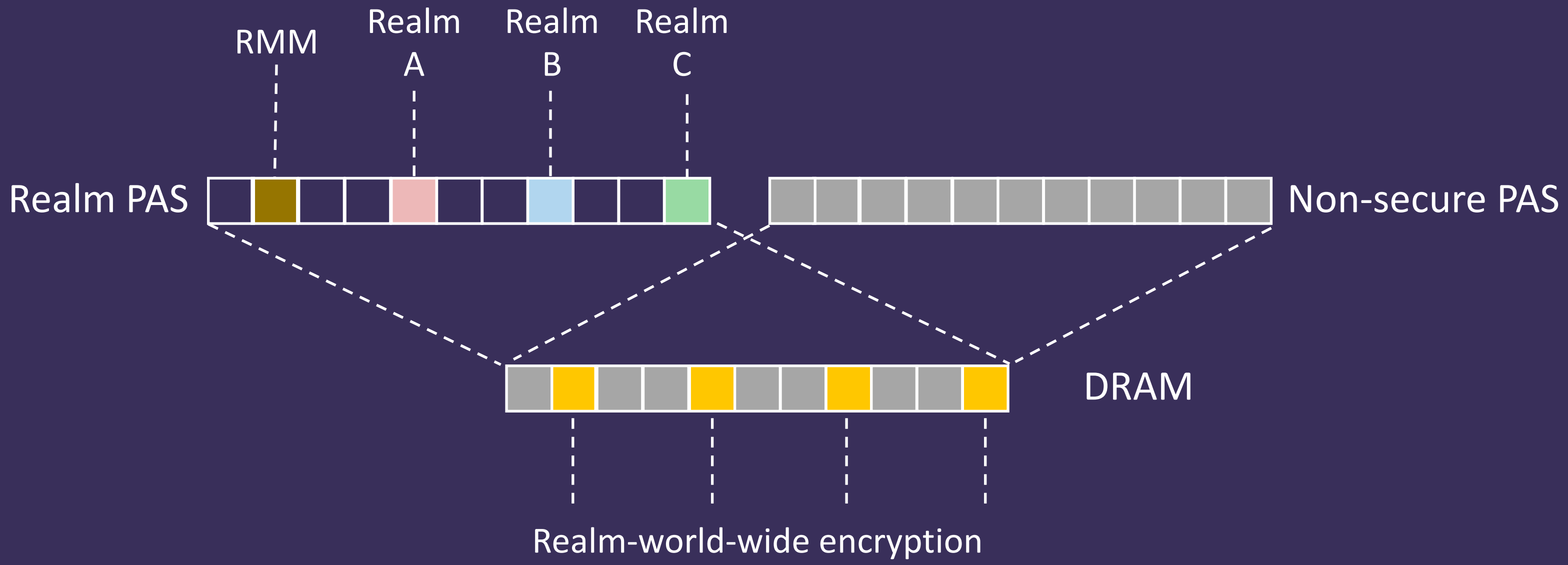
Granule protection



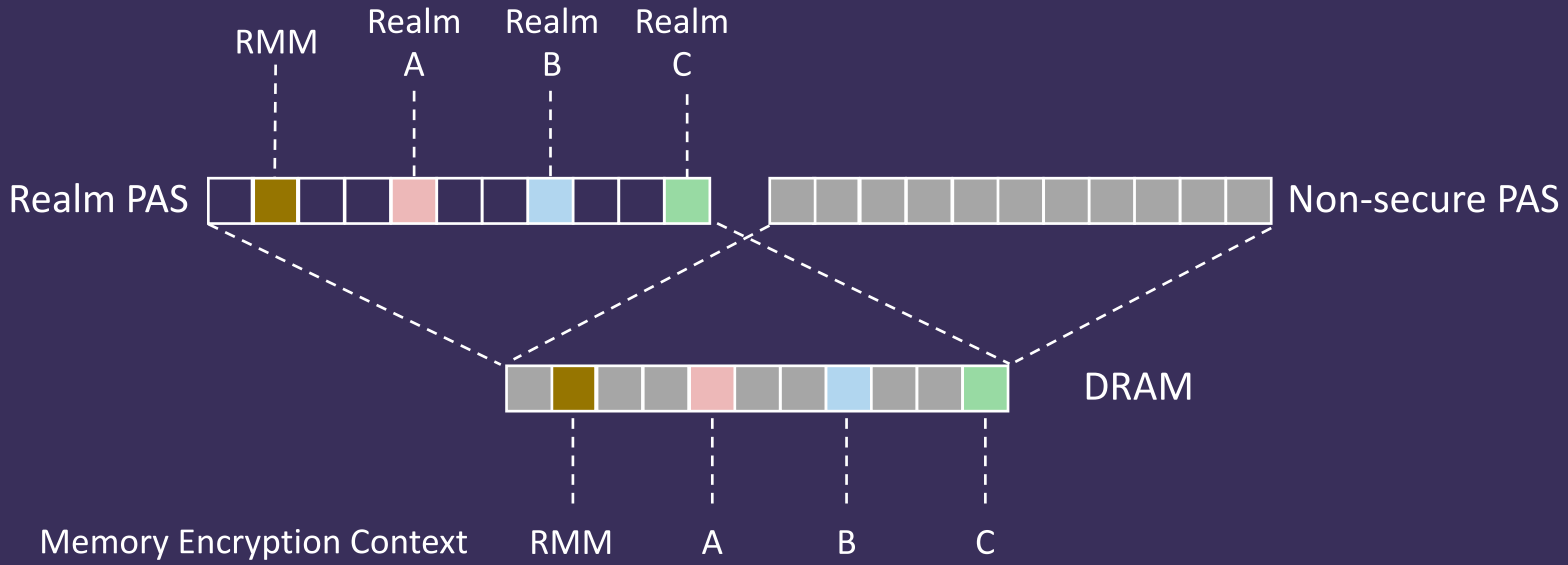
Granule protection



Granule protection



Memory Encryption Contexts (MEC)



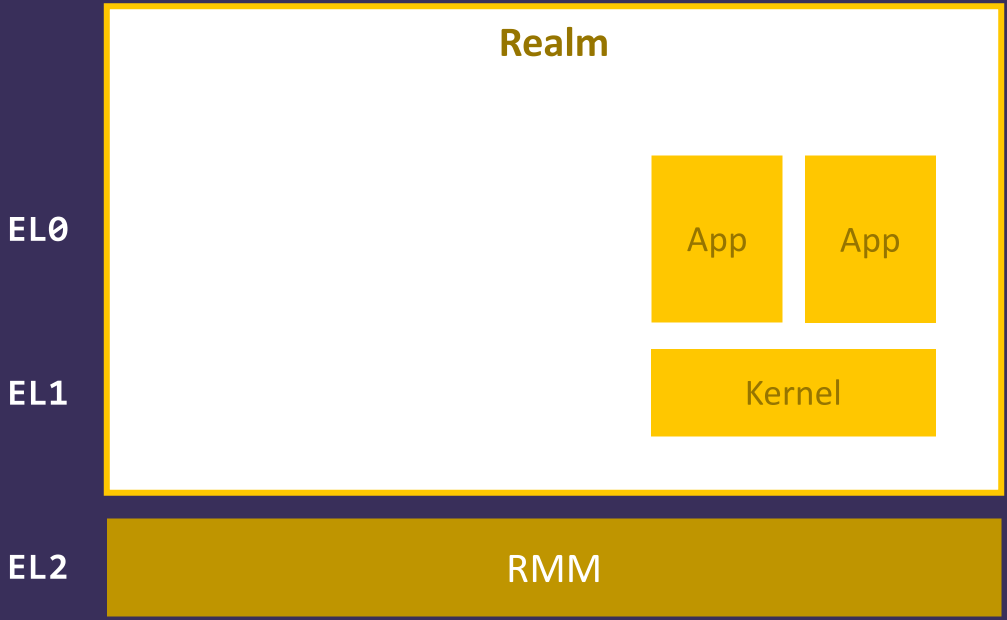
Evolution of Arm CCA

Further strengthen security guarantees provided to end users

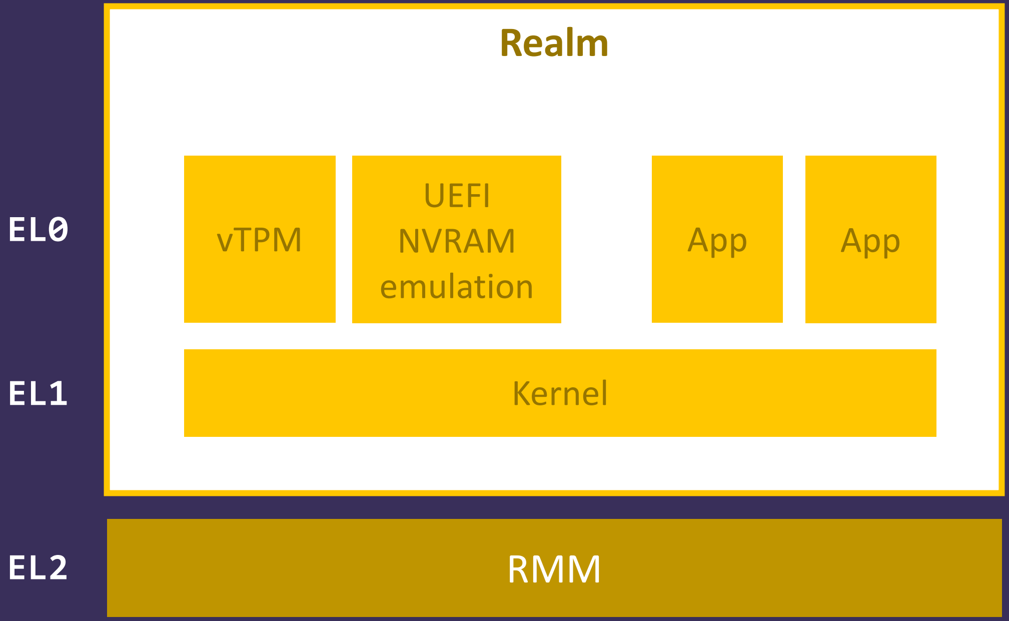
Provide feature parity between Realms and non-confidential VMs

Provide additional flexibility to Arm CCA platform owners

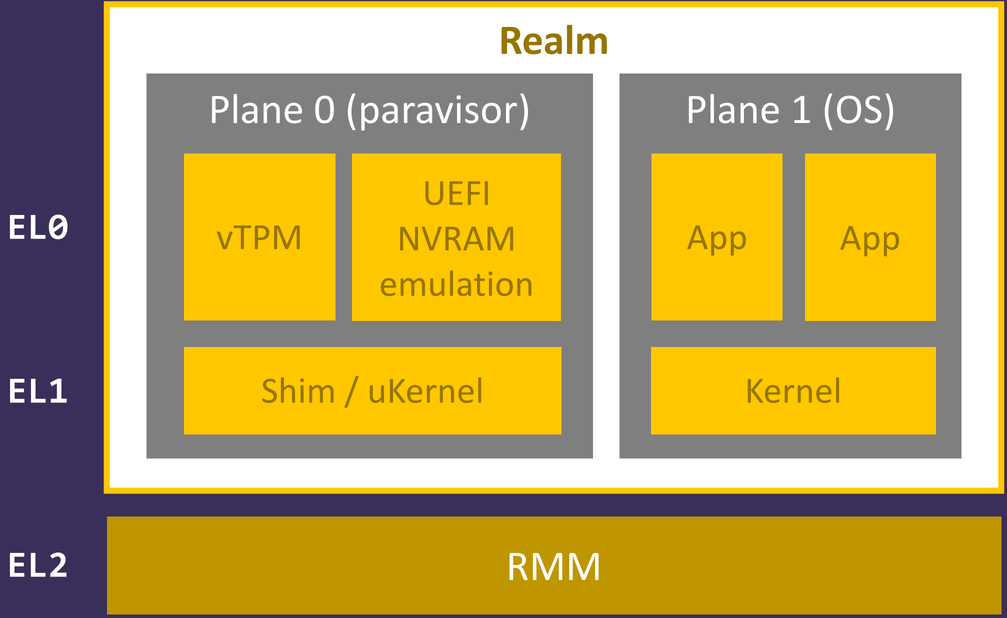
Intra-Realm privilege separation



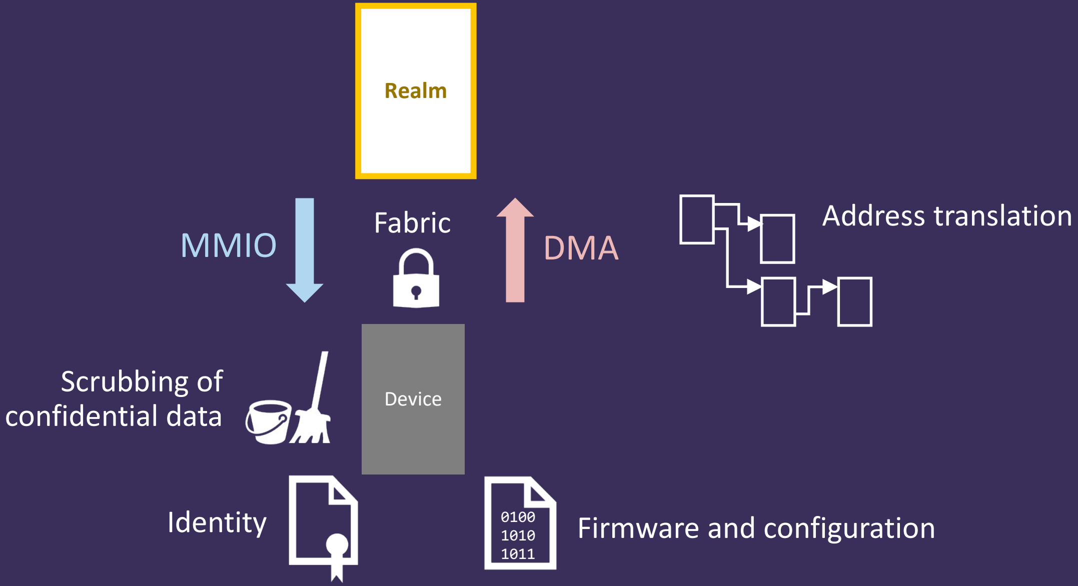
Intra-Realm privilege separation



Intra-Realm privilege separation



Device assignment



Device assignment

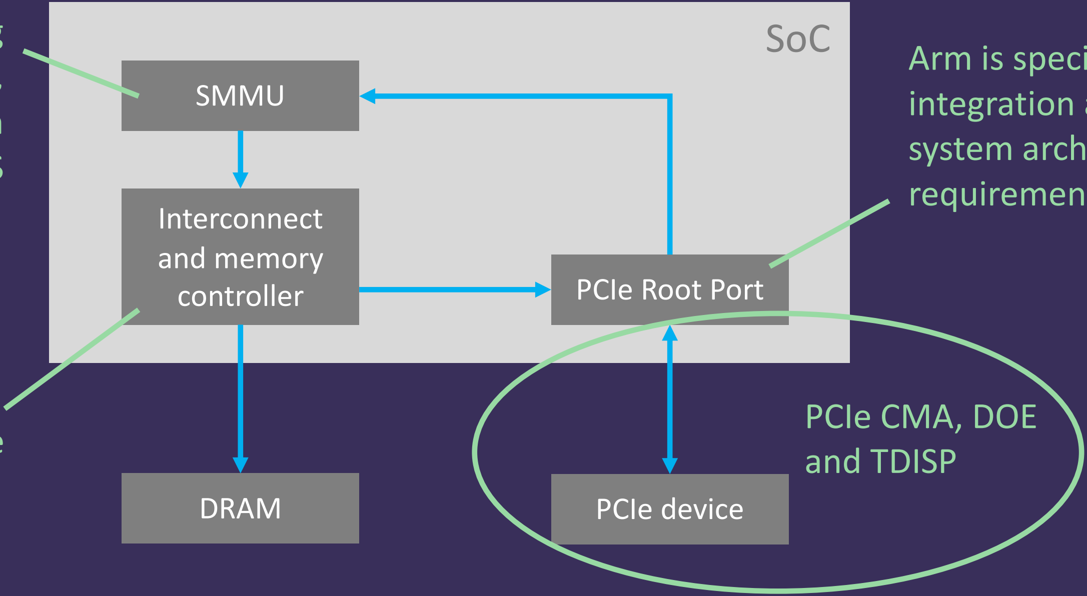
Trusted Device Interface Security Protocol (TDISP)

- A reference architecture which defines system components, and describes trust relationships
- Mechanisms to attest the identity and configuration of a device function
- A state machine, which ensures that
  - A device function can only access confidential data once it has been accepted by the VM
  - Confidential data is scrubbed from the device function before it it reassigned

Integrity and Data Encryption (IDE)

- Protection of the physical link between the host SoC and the assigned device function

Device assignment



RMM programming interface, and protection of Realm PAS

Already RME-aware

Arm is specifying integration and system architecture requirements

PCIe CMA, DOE and TDISP

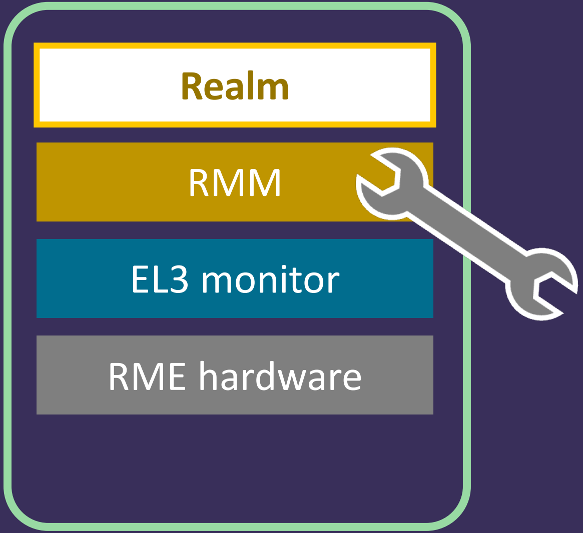
Evolution of Arm CCA

Further strengthen security guarantees provided to end users

Provide feature parity between Realms and non-confidential VMs

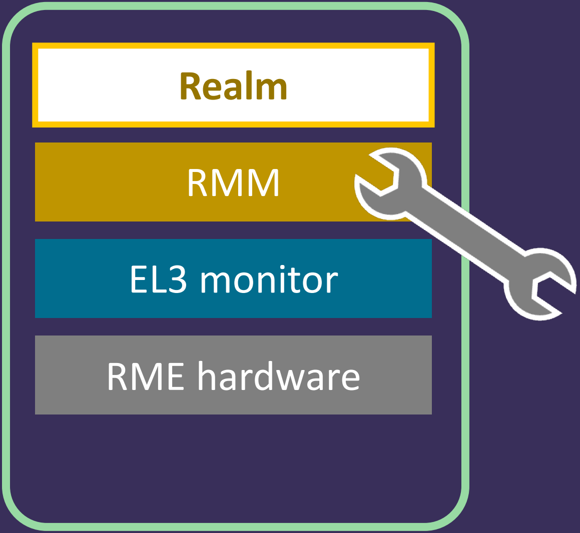
Provide additional flexibility to Arm CCA platform owners

CCA platform maintenance



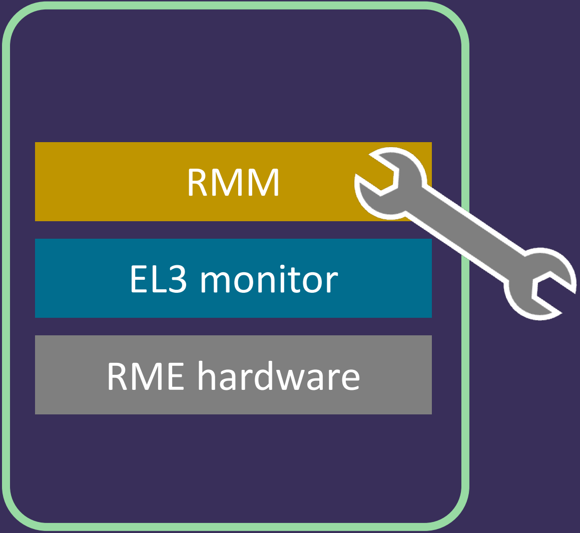
CCA platform maintenance

A. Tear down running workloads



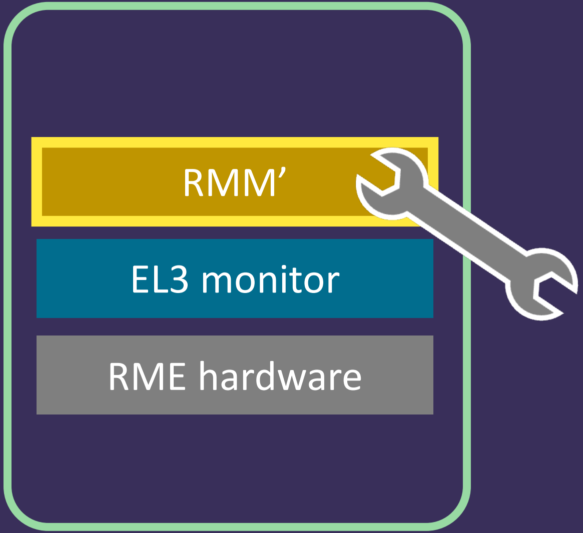
CCA platform maintenance

A. Tear down running workloads



CCA platform maintenance

A. Tear down running workloads



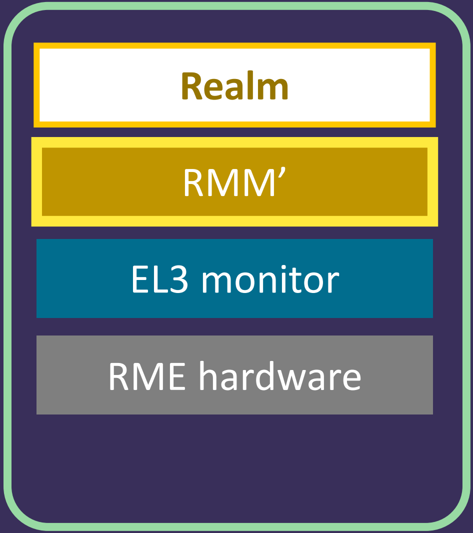
RMM’

EL3 monitor

RME hardware

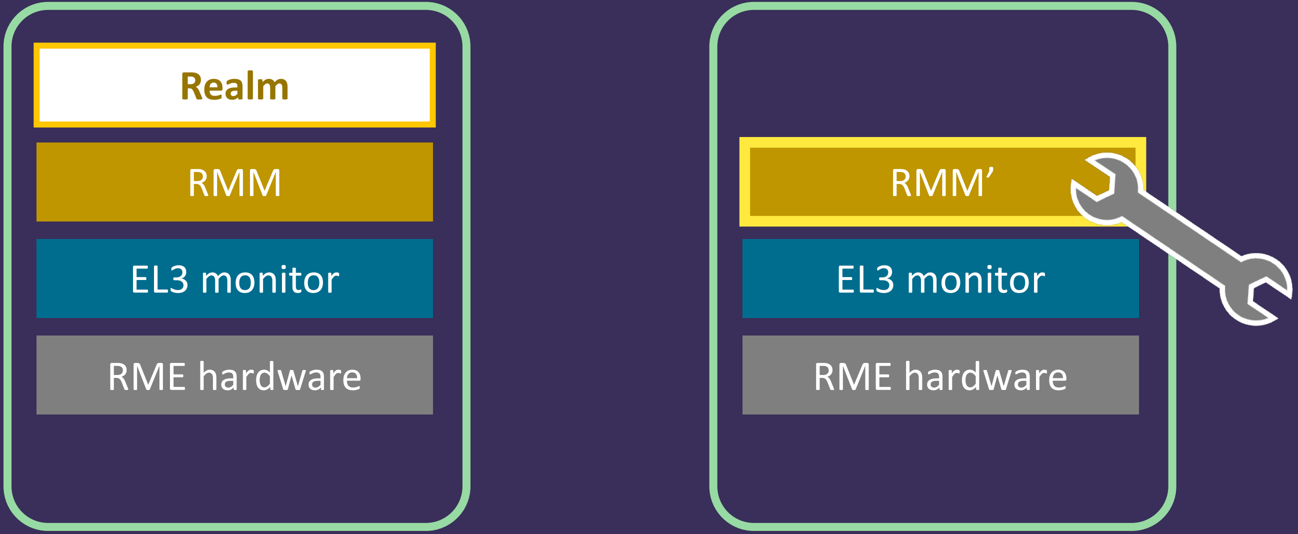
CCA platform maintenance

A. Tear down running workloads



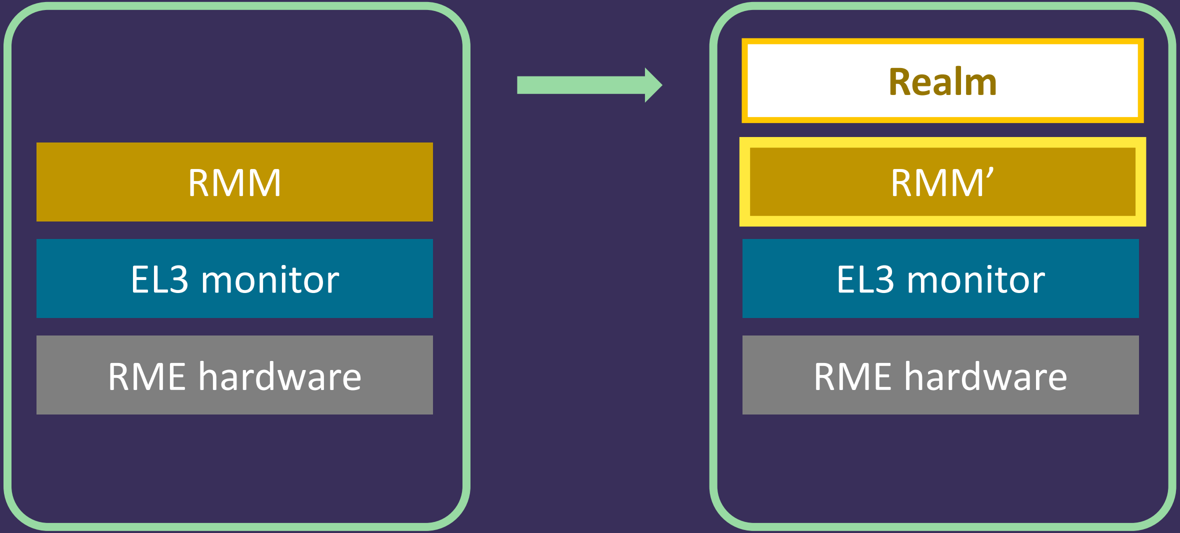
CCA platform maintenance

B. Migrate running workloads



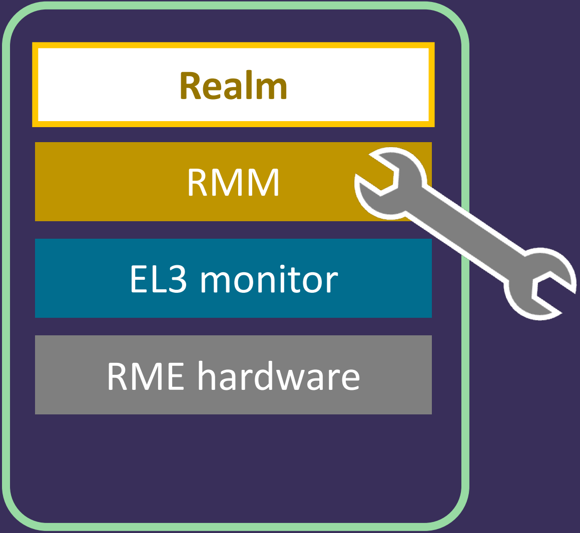
CCA platform maintenance

B. Migrate running workloads



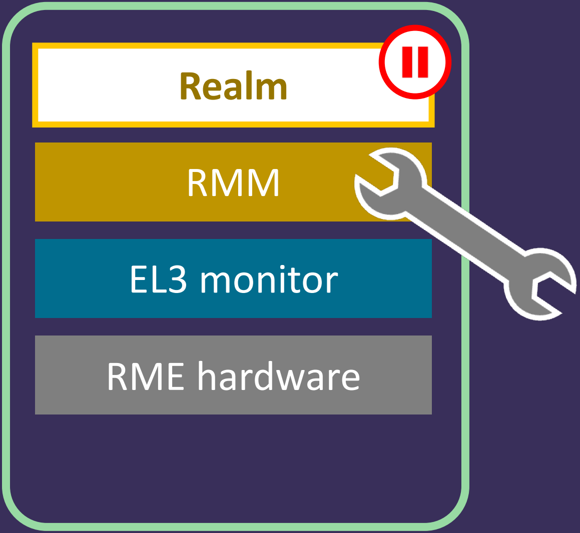
CCA platform maintenance

C. Update firmware underneath running workloads



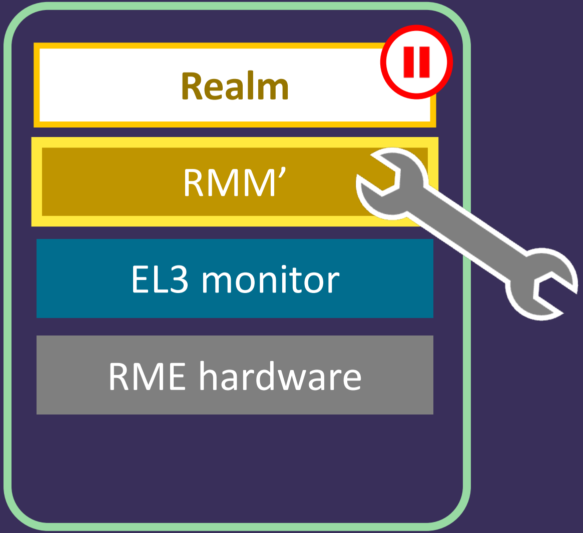
CCA platform maintenance

C. Update firmware underneath running workloads



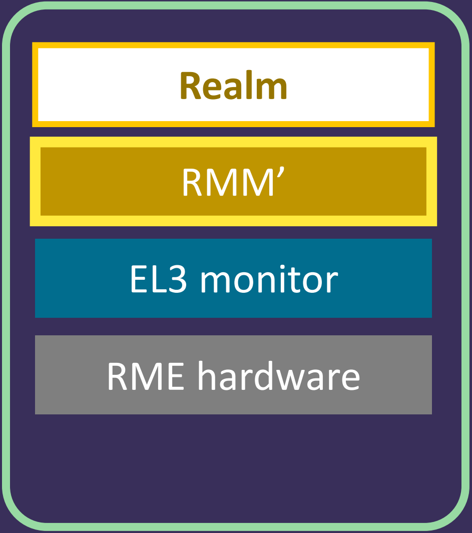
CCA platform maintenance

C. Update firmware underneath running workloads



CCA platform maintenance

C. Update firmware underneath running workloads



Realm

RMM’

EL3 monitor

RME hardware

Evolution of Arm CCA

Further strengthen security guarantees provided to end users

- Memory Encryption Contexts

Provide feature parity between Realms and non-confidential VMs

- Planes
- Device assignment

Provide additional flexibility to Arm CCA platform owners

- Live migration
- Live firmware activation

How Arm Is Supporting Ecosystem Developers

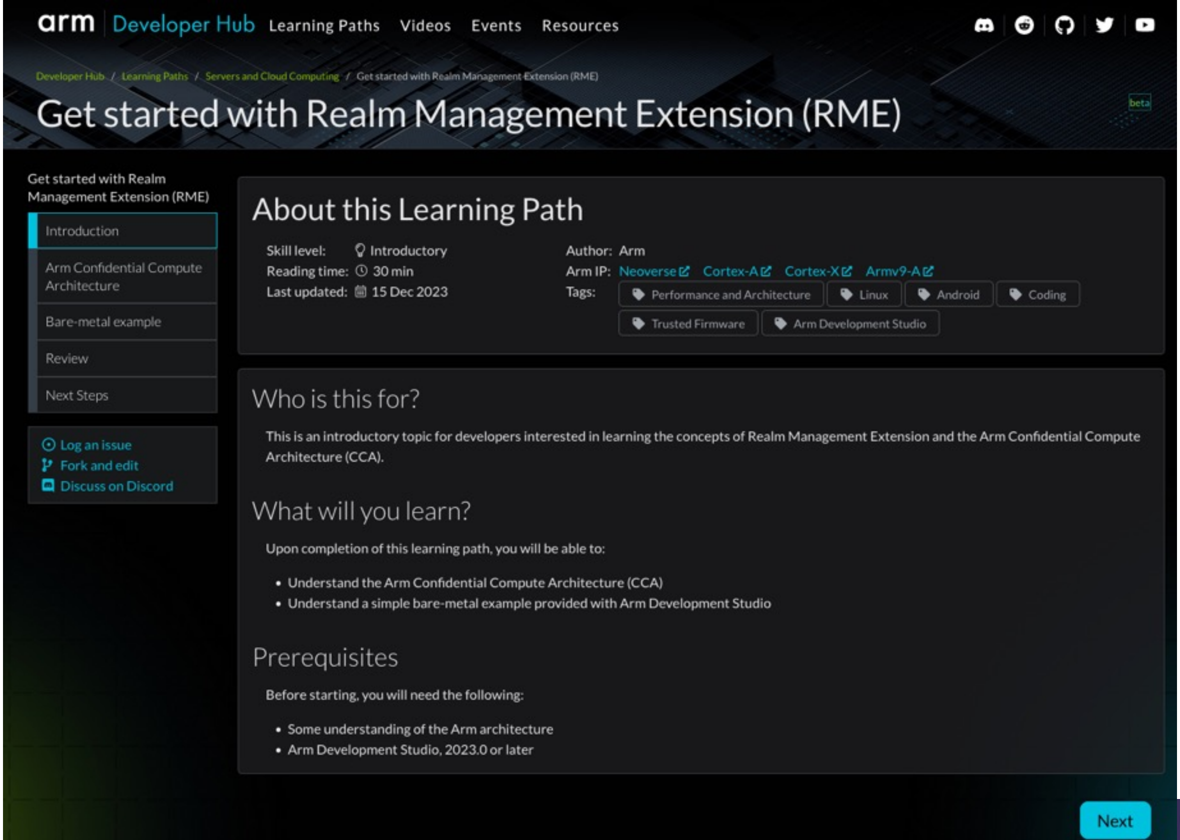
Nick Sample
Paul Howard

How Arm supports developers

- arm.com/developer-hub
- includes on-demand training events, blogs and videos
- Arm Developer Program: a community for peer-to-peer support
- learning paths

What is a learning path?

- visit learn.arm.com
- build your skills with step-by-step guidance on key tasks/workflows



Developer Hub / Learning Paths / Servers and Cloud Computing / Get started with Realm Management Extension (RME)

Get started with Realm Management Extension (RME)

Get started with Realm Management Extension (RME)

- Introduction
- Arm Confidential Compute Architecture
- Bare-metal example
- Review
- Next Steps

About this Learning Path

Skill level: Introductory
Reading time: 30 min
Last updated: 15 Dec 2023

Author: Arm
Arm IP: Neoverse Cortex-A Cortex-X Armv9-A
Tags: Performance and Architecture, Linux, Android, Coding, Trusted Firmware, Arm Development Studio

Who is this for?

This is an introductory topic for developers interested in learning the concepts of Realm Management Extension and the Arm Confidential Compute Architecture (CCA).

What will you learn?

Upon completion of this learning path, you will be able to:

- Understand the Arm Confidential Compute Architecture (CCA)
- Understand a simple bare-metal example provided with Arm Development Studio

Prerequisites

Before starting, you will need the following:

- Some understanding of the Arm architecture
- Arm Development Studio, 2023.0 or later

Learning paths on CC

- Get Started with Realm Management Extension (RME)
- Learn how to create a virtual machine in a Realm using Arm Confidential Compute Architecture (CCA)
- Run an application in a Realm using the Arm Confidential Compute Architecture (CCA)

…and more to follow

We want your feedback

- What skills or workflows connected with CC would you like guidance on?
- Complete our short questionnaire

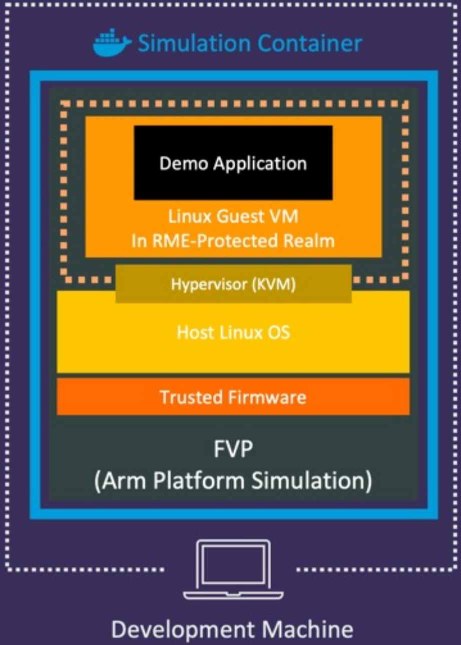
CCA Learning Experiences

The CCA Reference Stack

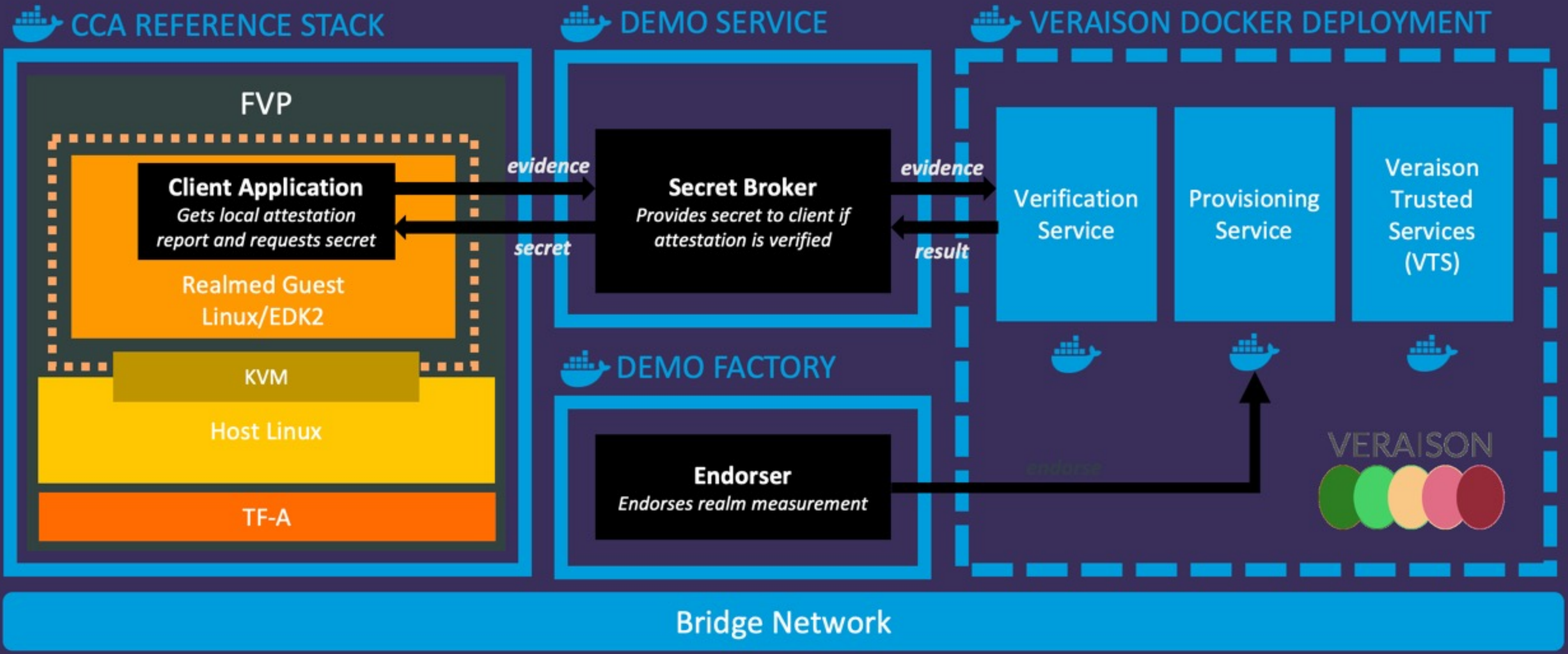
A Framework for Common Workflows

Attestation Flows and Patterns

The CCA Reference Stack



A Framework For Common Workflows



CCA REFERENCE STACK

FVP

Client Application
Gets local attestation report and requests secret

Realmed Guest Linux/EDK2

KVM

Host Linux

TF-A

evidence

secret

DEMO SERVICE

Secret Broker
Provides secret to client if attestation is verified

evidence

result

VERAISON DOCKER DEPLOYMENT

Verification Service

Provisioning Service

Veraison Trusted Services (VTS)

endorse

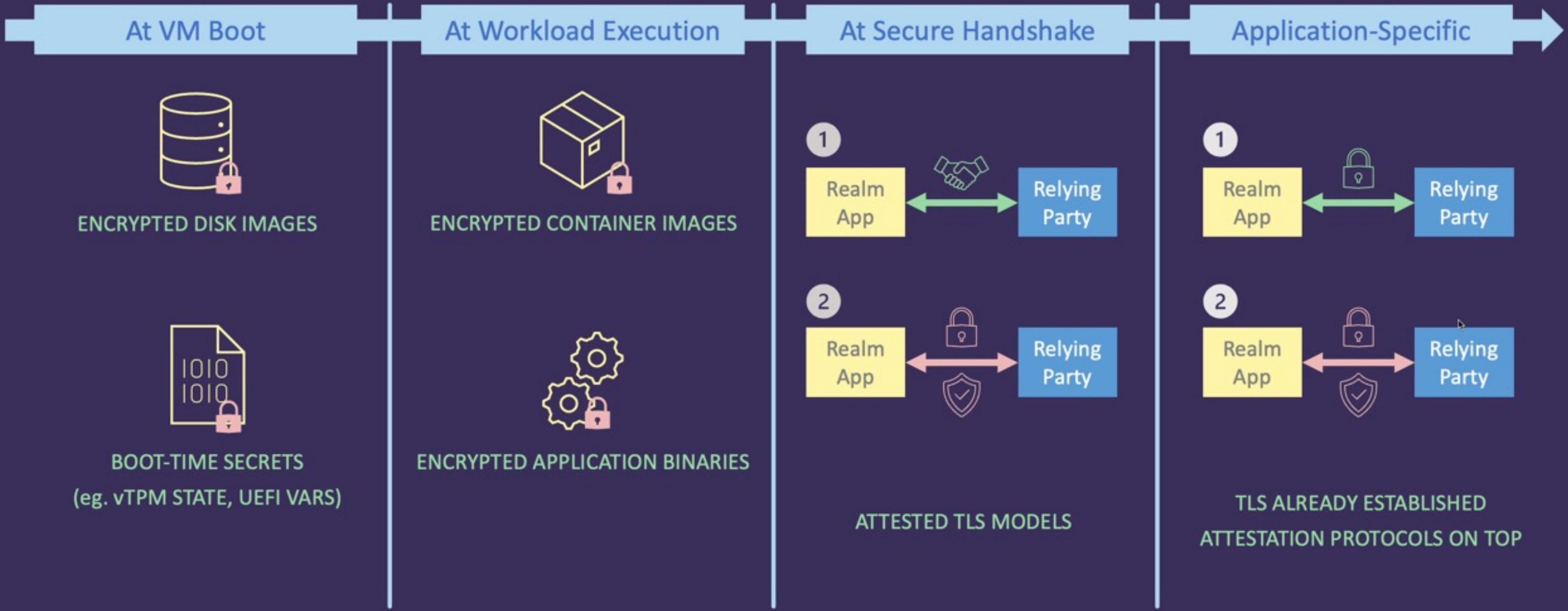
VERAISON

DEMO FACTORY

Endorser
Endorses realm measurement

Bridge Network

Attestation Flows and Patterns



At VM Boot

ENCRYPTED DISK IMAGES

BOOT-TIME SECRETS
(eg. vTPM STATE, UEFI VARS)

At Workload Execution

ENCRYPTED CONTAINER IMAGES

ENCRYPTED APPLICATION BINARIES

At Secure Handshake

ATTESTED TLS MODELS

Application-Specific

TLS ALREADY ESTABLISHED
ATTESTATION PROTOCOLS ON TOP

We want your feedback

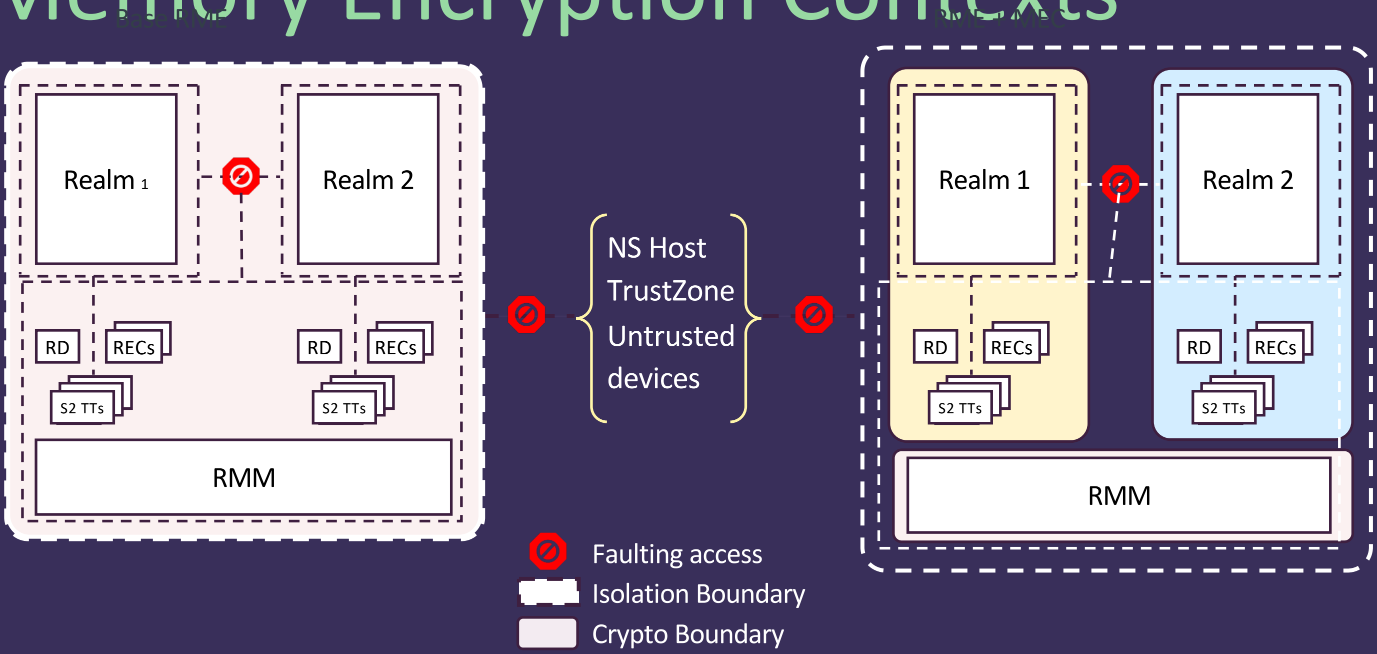
- What skills or workflows connected with CC would you like guidance on?
- Complete our short questionnaire

Contact us!

info@oc3.dev

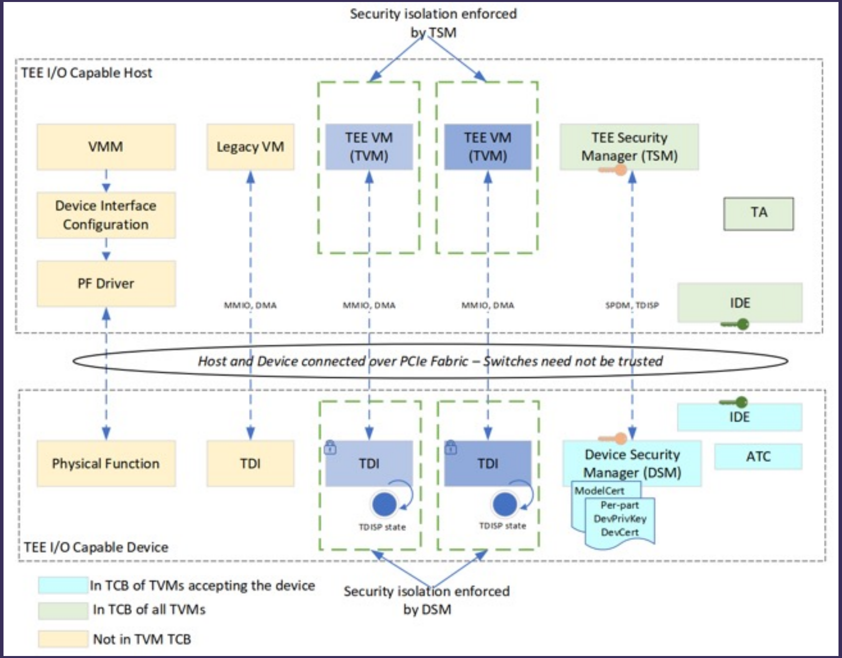
Backup slides

Memory Encryption Contexts



Device assignment

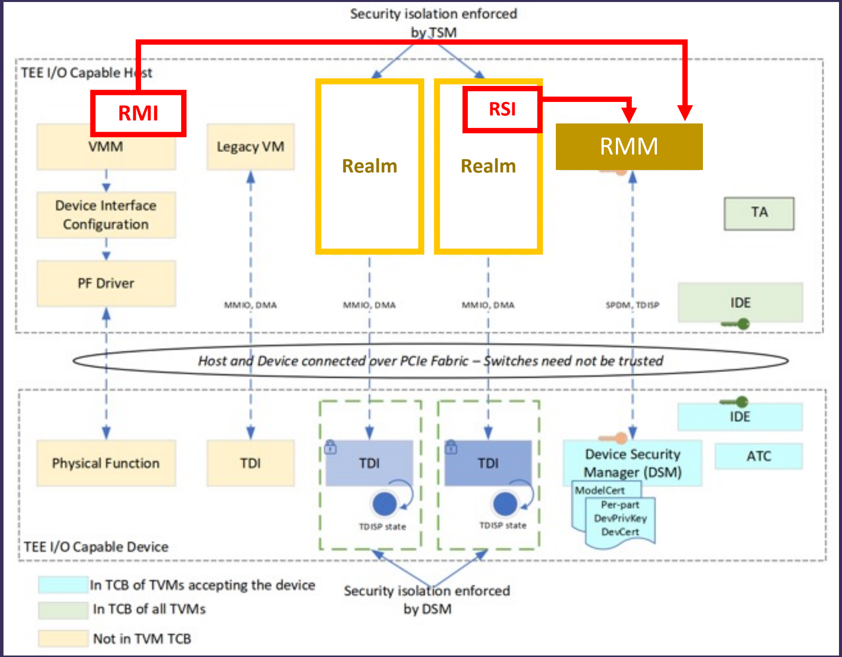
PCIe TDISP reference architecture



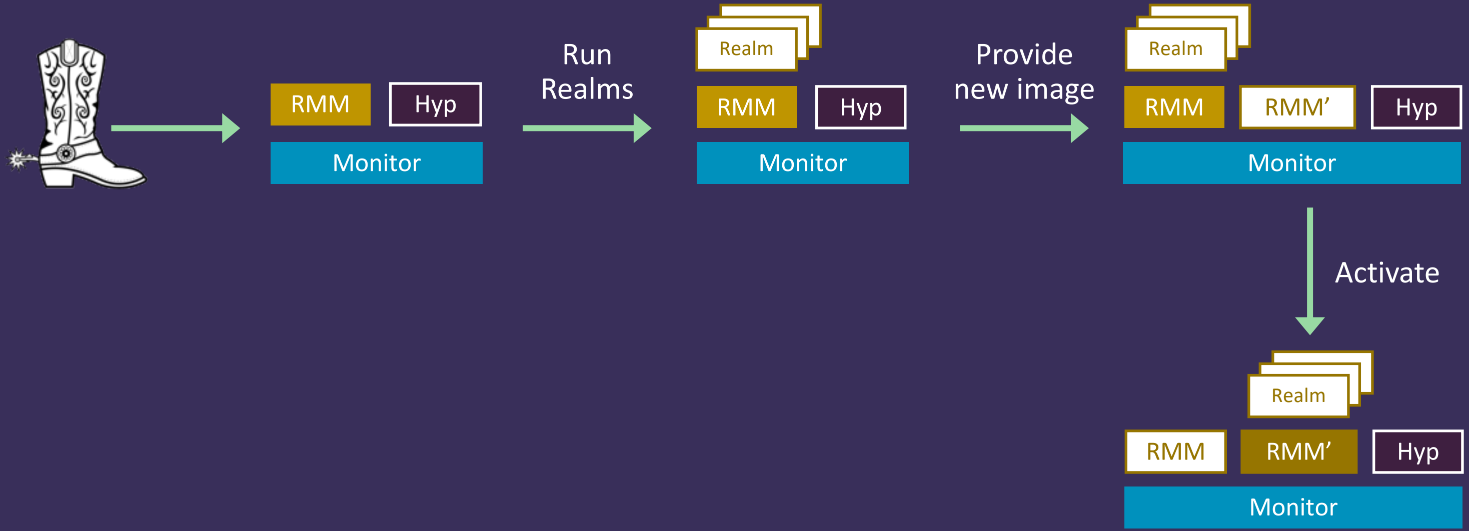
Security isolation enforced by TSM
TEE I/O Capable Host
VMM
Device Interface Configuration
PF Driver
Legacy VM
TEE VM (TVM)
TEE VM (TVM)
TEE Security Manager (TSM)
TA
IDE
MMIO, DMA
MMIO, DMA
MMIO, DMA
SPDM, TDISP
Host and Device connected over PCIe Fabric – Switches need not be trusted
IDE
Physical Function
TDI
TDI
TDI
Device Security Manager (DSM)
ATC
ModelCert
Per-part DevPrivKey DevCert
TDISP state
TDISP state
TEE I/O Capable Device
Security isolation enforced by DSM
In TCB of TVMs accepting the device
In TCB of all TVMs
Not in TVM TCB

Device assignment

PCIe TDISP reference architecture


Security isolation enforced by TSM
TEE I/O Capable Host
RMI
VMM
Device Interface Configuration
PF Driver
Legacy VM
Realm
Realm
RSI
RMM
TA
IDE
MMIO, DMA
MMIO, DMA
MMIO, DMA
SPDM, TDISP
Host and Device connected over PCIe Fabric – Switches need not be trusted
IDE
Physical Function
TDI
TDI
TDI
TDISP state
TDISP state
Device Security Manager (DSM)
ATC
ModelCert
Per-part DevPrivKey DevCert
TEE I/O Capable Device
In TCB of TVMs accepting the device
In TCB of all TVMs
Not in TVM TCB
Security isolation enforced by DSM

Live firmware activation


RMM
Hyp
Monitor
Run
Realms
Realm
RMM
Hyp
Monitor
Provide
new image
Realm
RMM
RMM'
Hyp
Monitor
Activate
Realm
RMM
RMM'
Hyp
Monitor